

PETSc

Solving Linear and Nonlinear PDES

William Gropp



What is PETSc?

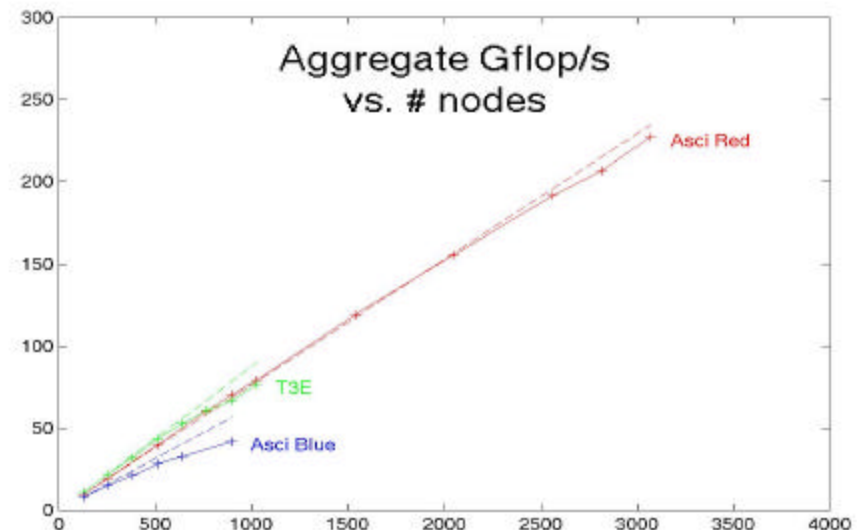
- PETSc is a library of routines for solving linear and nonlinear systems of equations that arise from discretizing partial differential equations
- PETSc can be used by programs written in C, Fortran, or C++
 - ◆ PETSc does not take over your program

Why use PETSc?

- PETSc provides access to the best numerical algorithms
 - ♦ A wide range of preconditioned linear solvers and nonlinear equation methods
- PETSc handles nearly all of the parallel computing issues
 - ♦ no MPI calls in your program!
- PETSc provides more than just nonlinear solvers
 - ♦ Support for constructing sparse matrices
 - ♦ Support for solution output, including parallel NetCDF

How Fast is PETSc?

- Fun3d, a legacy Fortran application, parallelized with PETSc, won a 1999 Gordon Bell prize for achieving 220 Gflops on a fully unstructured g exterior CFD problem



Where is PETSc Available?

- Nearly everywhere!
 - ♦ Portable to a wide range of systems, including most Unix (including Mac OS/X) and Windows
 - ♦ Runs on laptops
 - Does not require MPI, though MPI runs on laptops too
 - ♦ Installed at NCSA
 - ♦ BSD-like license (free to all users, even for commercial use)
 - ♦ Open source
 - ♦ Support at petsc-maint@mcs.anl.gov

What's the Catch?

- PETSc not suitable for all programs
 - ♦ Single component linear systems often show little benefit in *performance*
- PETSc is large and can be daunting
 - ♦ 150+ page manual
 - ♦ Steep learning curve
- But
 - ♦ Many tutorial examples included
 - Heavily commented code in C, Fortran 77, and Fortran 90
 - ♦ This workshop!

Basic Linear Solver Code

```

SLES  sles;          /* linear solver context */
Mat    A;            /* matrix */
Vec    x, b;         /* solution, RHS vectors */
int     n, its;       /* problem dimension, number of iterations */

MatCreate(PETSC_COMM_WORLD,PETSC_DECIDE,PETSC_DECIDE,n,n,&A);
MatSetFromOptions(A);
/* (code to assemble matrix not shown) */
VecCreate(PETSC_COMM_WORLD,&x);
VecSetSizes(x,PETSC_DECIDE, n);
VecSetFromOptions(x);
VecDuplicate(x,&b);
/* (code to assemble RHS vector not shown)*/

SLESCreate(PETSC_COMM_WORLD,&sles);
SLESSetOperators(sles,A,A,DIFFERENT_NONZERO_PATTERN);
SLESSetFromOptions(sles);
SLESSolve(sles,b,x,&its);
SLESDestroy(sles);

```